

# Fast image colourisation using the isolines concept

Adam Popowicz<sup>1</sup> · Bogdan Smolka<sup>1</sup>

Received: 8 December 2015 / Revised: 11 July 2016 / Accepted: 22 August 2016 /

Published online: 7 September 2016

© The Author(s) 2016. This article is published with open access at Springerlink.com

**Abstract** Colourisation is the process of adding colour information to monochromatic images and video sequences. Generally, colourised images are visually more attractive and they are advantageous in areas such as medicine, astronomy, microscopy or education, in which the visualisation of image information plays a crucial role. In this paper, we introduce a novel concept of grayscale image colourisation, which is based on the idea of isolines. The algorithm is straightforward, easy to implement, computationally efficient and it allows a very high colourisation quality to be obtained. The proposed approach utilises the distance transform applied to a grayscale digital image. The final colourisation is performed via the weighted blending of scribbled colours. The comparison with similar state-of-the-art colourisation techniques that is presented proves that the method is computationally more efficient while producing outputs that have the same high quality. Additionally, we provide the pseudocode for the easier implementation of the algorithm and show its multidisciplinary applications.

**Keywords** Image processing · Colourisation · Visualisation

## 1 Introduction

Image colourisation is a method of transforming a grayscale image into a colourised one. The intensity of each pixel is replaced by a colour representation: R, G and B channels or, as in YCrCb, two chrominance and one luminance channels. This task is not trivial and there are no explicit solutions for transforming a one-dimensional pixel value (intensity) into a three dimensional vector. Thus, such a process can be performed in various ways, which results in a wide range of developed colourisation approaches.

---

✉ Adam Popowicz  
apopowicz@polsl.pl

<sup>1</sup> Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland

The first colourisation algorithms to enhance still images and video sequences were applied in 1984 [9]. Nowadays, with the increasing efficiency of modern computers, the methods are much more sophisticated and they make use of the computational power that is available. The algorithms, in general, can be applied to two methods: automatic and semi-automatic. The first one includes the algorithms that compare the grayscale image with a given colour reference. Using various approaches, they try to fit the parts of both images and to properly estimate the colour of each image region. In the second method, the user indicates the colour hints in the form of scribbled strokes of a given colour (the so-called scribbles). This way the colourisation process starts from the inserted scribbles and expands, to finally fill up the whole image.

While the quality of the colourisation outputs is currently at a very high level, which makes the results almost indistinguishable from the original colour images, the complexity of the methods has also increased. It should be noted, that the colourisation that employs the semi-automatic approach is an interactive process, in which a user indicates scribbles, observes the outputs and modifies or inserts additional scribbles in order to achieve the desired final effect. Hence, the computational efficiency should also be considered and maximised in order to reduce the time a user spends on colourisation. The simplicity of the implementation, as well as the reduction of the computational burden, while preserving a high output quality were, therefore, the main goals of the new isoline-based colourisation technique that is presented in this article.

The article is organised as follows. In Section 2, we present the range of possible colourisation algorithms that are available in the literature. In the next section, we introduce the basic idea of our method and present some modifications that lead to its better performance and lower computational complexity. In Section 4, we investigate the quality issues of colourisation, which include parameter optimisation and a performance comparison with other colourisation methods. In Section 5, we present an analysis of the computational complexity of the new algorithm as compared with some reference techniques. Finally, in the last two Sections, we present some possible applications of the proposed colourisation method and some final conclusions.

## 2 State-of-the-art colourisation methods

As mentioned in the introduction, there are two main types of colourisation methods: automatic and semi-automatic. The first requires the reference colour image, while the second one needs human interaction. In this brief literature review we enumerate some algorithms of both colourisation approaches.

The authors of [19] developed the seminal work on automatic image colourisation. They suggested employing the idea of colour transfer between images presented in [13]. The authors make use of  $\alpha\beta$  colour model [16] instead of the classic RGB space. They investigate the statistics of pixel intensities in a sliding processing window utilising their standard deviation and mean value. This way they transfer the colours to the parts of grayscale image which show similar luminance statistics. An exemplary application of the algorithm for biomedical image colourisation is presented in [23]. The authors add some modifications to the original idea so that the measure of similarity between image patches is tuned by a single parameter.

In [24] another method of automatic image colourisation using a reference image is presented. The authors suggest using the pattern continuity and spatial consistency approach instead of analyzing the image luminance properties. First, the image is segmented and then,

using Gabor wavelet analysis, the vector of features used as a training set in the classification algorithm is calculated.

Another proposition of automatic approach is dedicated to the colourisation of infrared videos [20]. The authors suggest obtaining reference colour images from flash assisted photos taken during infrared video acquisition. They apply the colour transfer from the references to the whole sets of images between the so-called key frames in a video shot and utilise the luminance distance between the reference and given infrared image in each local window.

The semi-automatic methods, in contrast to the automatic ones, do require some additional user effort. It usually includes inserting the hints in a form of colour scribbles which indicate the regions of a given colour and are the starting points for the colour flooding performed by the algorithms.

An example of semi-automatic colourisation is presented in [7, 8]. The authors investigate the possibility of using the minimisation of colour difference between the adjacent pixels of similar intensity. The algorithm starts from the colour scribbles indicated by the user and expands over the whole image. The minimisation can be achieved by typical approaches, like the Least Square algorithm. The authors also show the ability of their algorithm to colourise the video sequences which is simply an extension to the spatio-temporal domain.

Another method of scribbled-based colourisation is described in [21]. In this method the authors investigate the digital paths between colour scribbles and grayscale image pixels. They use Dijkstra's algorithm to find the most homogeneous (in terms of the intensity gradients) digital paths. The final colour of a pixel is calculated as a weighted average of all assigned colours, where the optimal weights are inversely proportional to the costs of the optimal paths.

In [6] the use of morphological distance transformation for the colourisation purposes is presented. The authors, like in [21], investigate the digital paths and the corresponding gradients. However, they use the estimation of the shortest paths provided by the so-called double-scan algorithm [14], in which the computation time is much lower than that of Dijkstra's algorithm. The cost of the digital paths is calculated using squared intensity differences between adjacent pixels and considering an additional penalty for the path length.

The utilisation of distance transformation for the colourisation is also presented in [22]. The authors apply fuzzy clustering of a grayscale image using inserted scribbles as initial cluster centers. They make use of the estimation of Euclidean distance, provided by the distance transformation, to find the fuzzy membership of the pixels to each of the scribbles and then they perform weighted colourisation.

The use of local Markov properties of the image is provided in [1]. The colours are transferred from the scribbles to the neighboring pixels which minimise the difference in RGB space. This method represents the concept of seed pixels propagation. In [2] the image pixels are divided into unique regions using the watershed segmentation [11]. The initial oversegmentation is removed by region flooding, so that the pixels are being merged as long as the number of segments is equal to the number of colour scribbles indicated by a user.

Although most of the algorithms use pixel intensity analysis, articles [3–5] present the idea of utilising the textural properties of an image. The authors use two types of distance metrics: the intensity distance, which promotes smooth digital paths, and the textural distance, which privileges sharp intensity transitions. This way, the algorithms are able to properly colourise the parts of the image containing highly rugged structures.

### 3 Isoline based image colourisation

#### 3.1 Basic concept

Our method utilises the semi-automatic colourisation approach which, like in similar methods, requires scribbles to be inserted into the grayscale image. The goal of the main part of our algorithm is to properly assign the degree of membership to a given colour/scribble to the pixels. Our idea is very similar to that of the isolines on the geographic maps. To do this, we look for the digital path from a scribble to the pixel that minimises the maximum intensity deviation. We calculate the deviation as the maximum difference between the intensity of a scribbled pixel and the intensity of any of the pixels on the digital path. The maximum intensity deviation  $D_{i,j}$  between the pixels at positions  $i$  and  $j$  is defined as:

$$D_{i,j} = \min_{\Phi} \{ \max |I_k - I_i|, k \in \phi_{i,j} \}, \quad (1)$$

where  $I_i$  and  $I_k$  are the intensities of the pixels at  $i$  and  $k$ ,  $\phi_{i,j}$  is the set of adjacent pixel positions that form a particular path between the pixels at  $i$  and  $j$ ,  $\Phi$  is the set of all the paths connecting the pixels at  $i$  and  $j$ .

In this way, we create the distance map (assignment map) for each pixel, that is very similar to a geographic representation with isolines depicting the points with the same properties, such as height or temperature. In our case the isolines represent the pixels with the same value of intensity deviation encountered on the path that leads to a given scribble. In Fig. 1 we show the isoline representation of the introduced distance map for a scribble that is indicated on an exemplary grayscale image.

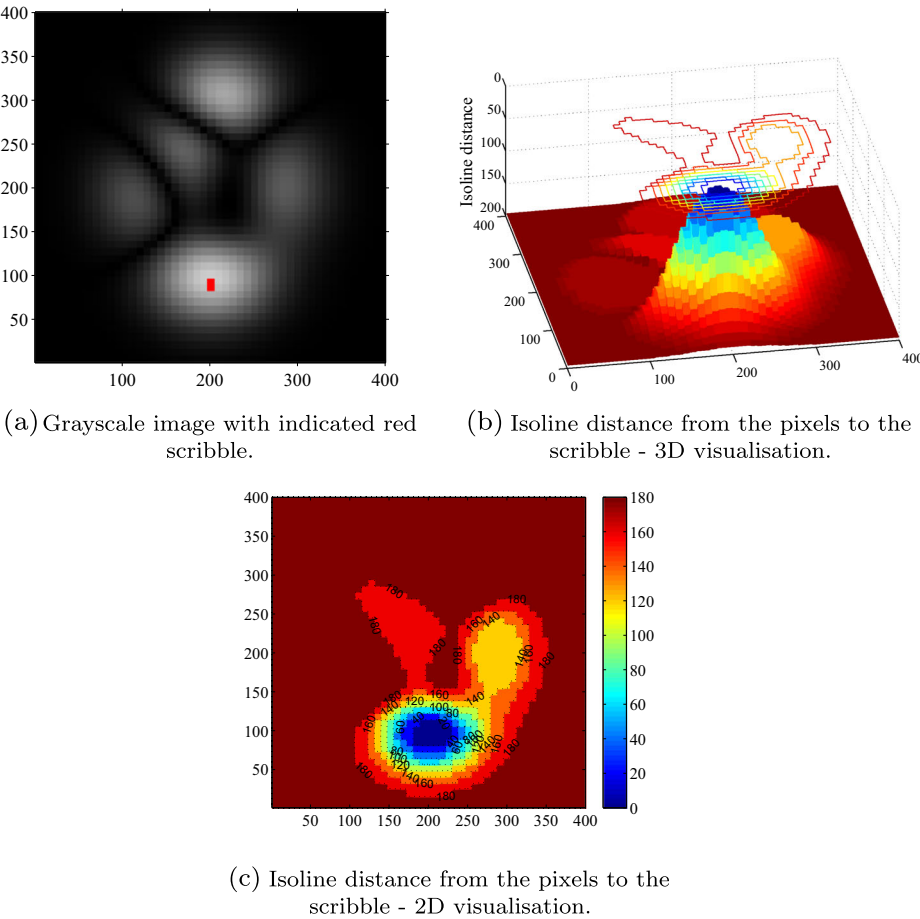
In Fig. 2, we present the principle of our idea in a one-dimensional case. In this example, only one pixel, which is indicated in red, is scribbled. We should note that the isoline distance cannot decrease. For the last pixel (intensity 9), the isoline distance equals 3, although the intensity difference from the scribble (seed) is only 1.

In two-dimensional space the problem is more complicated because of the many possible paths that start from the scribble pixels. Moreover, one scribble usually consists of a group of pixels. Thus, the isoline calculation should involve a search among all of the possible paths from all of the pixels included in a given scribble. Figure 3 shows an example of a simple 2D isoline distance calculation using 4-neighborhood relation. Although computation on a 2D image can be executed using Dijkstra's algorithm, this requires a great deal of computational time and therefore we rejected such a solution. Instead, we developed an algorithm that is based on the local estimation of the introduced isoline distance.

Our algorithm includes a modification of the double-scan method, which is exploited in the topographic distance transformations [6, 15]. This distance transformation allows the Euclidean distance to be estimated using very simple and computationally efficient operations in a local moving window. It employs the double scan algorithm, which considers only the information from the three neighbouring pixels whose selection depends on the direction of the scan (left-to-right, top-to-bottom or right-to-left, bottom-to-top, see Fig. 4) at each step. While the pixels initially have an infinite distance, only those within the scribbles are initialised with zeros (see Fig. 5 b). Then, at each step, the new distance value is calculated using the distances of the pixels within the window:

$$D_1 \leftarrow \min \{D_1, D_2 + 1, D_3 + 1\}, \quad (2)$$

where  $D_1$ ,  $D_2$  and  $D_3$  are the distance values of the corresponding pixels  $P_1$ ,  $P_2$  and  $P_3$ . Figure 5 presents an example of classic distance calculations using the double-scan algorithm.



**Fig. 1** Isoline distance visualisation

Since the main advantage of the proposed algorithm is its high computational efficiency, the double scan was selected rather than Dijkstra’s algorithm. The predominance of the double-scan algorithm can be explained as follows. Assuming that the image matrix consists of  $M \times N$  pixels, a connected graph of the pixel grid has  $E = N(M - 1) + M(N - 1)$  edges and  $V = MN$  nodes (no diagonal connections). The number of operations for calculating distance using Dijkstra’s algorithm ( $O_{Dij}$ ) can be optimised by Fibonacci Heaps [10] to:

$$O_{Dij} = E + V \log V = 2V - M - N + V \log V.$$

(3)

Intensity	10	11	12	11	13	12	11	9
Isoline distance	0	1	2	2	3	3	3	3

**Fig. 2** One-dimensional isoline distance calculations

10	11	12	13	13
9	17	12	18	15
11	15	13	11	17
12	13	14	13	12
13	14	15	14	13

(a) Exemplary intensity array with path (gray) of the lowest cost that joins the pixels indicated in red and green.

Intensity	10	11	12	12	13	11	13	12	13
Isoline distance	0	1	2	2	3	3	3	3	3

(b) Isoline distance for the path indicated in (a).

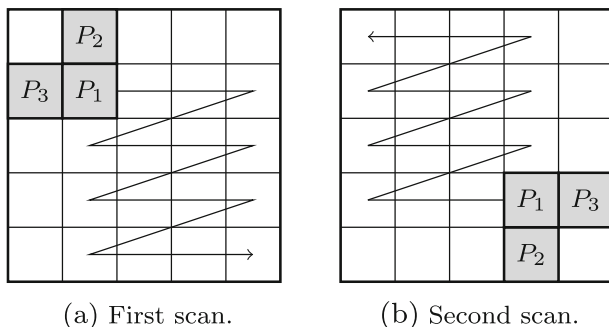
**Fig. 3** Two-dimensional isoline distance calculation. Best path - gray, seed pixel - red, destination pixel - green

For the double-scan algorithm, this number is lower and equals:

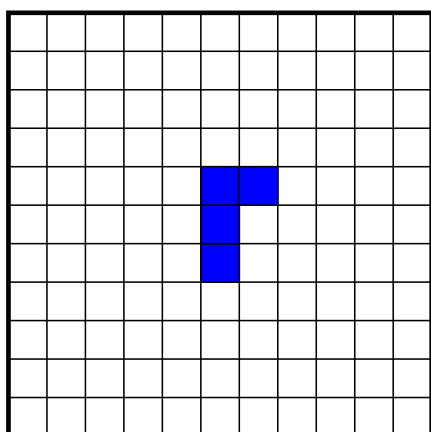
$$O_{DS} = 2(M - 1)(N - 1) = 2V - 2M - 2N + 2. \quad (4)$$

The dependencies of the number of operations required by Dijkstra's algorithm and double-scan algorithm, on the number of pixels, are illustrated in Fig. 6. The number of pixels is selected so that it corresponds to the range of images:  $100 \times 100$  to  $400 \times 400$  pixels. For such a range, both dependencies are linear and the double-scan is approximately 3.6 times faster, which was also confirmed by the results of the evaluations presented in Section 5.

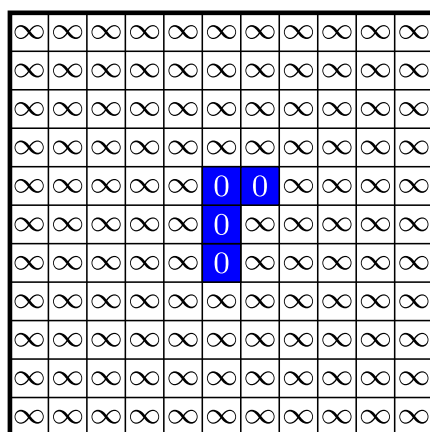
In our algorithm, we perform a series of operations using the pixels  $P_1$ ,  $P_2$  and  $P_3$  to obtain one distance map for every colour scribble indicated in the image. First, we initialise the isolines distance array, denoted as  $\mathcal{J}$ , so that each pixel that is included in the scribbles is given a 0 isoline distance, while the others are set to  $\infty$ . We also establish an auxiliary *propagated seed intensity* (PSI) array, which is denoted as  $S$ , wherein only the seed pixels indicated with a given colour are initially given their grayscale value while others are set to  $\infty$ . The propagated seed intensity array is crucial in our algorithm since it enables intensities from the seed (scribbled) pixels to be spread, so that eventually each pixel is given the propagated seed intensity that is equal to the intensity of the closest scribbled pixel.



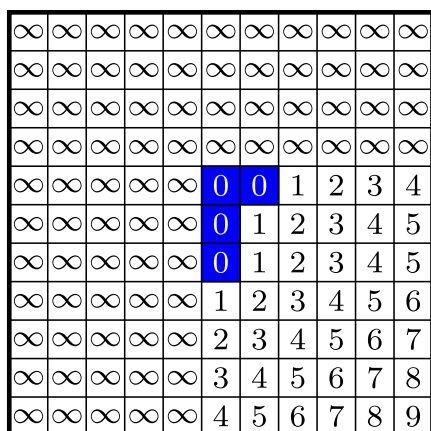
**Fig. 4** Double scan algorithm - moving window



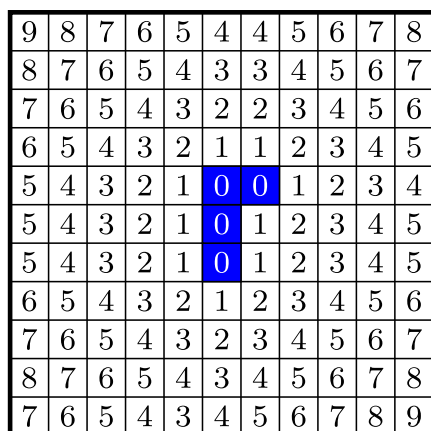
(a) Indicated scribble.



(b) Distance initialization.



(c) First scan result.



(d) Second scan result.

**Fig. 5** Example of a double-scan distance calculation

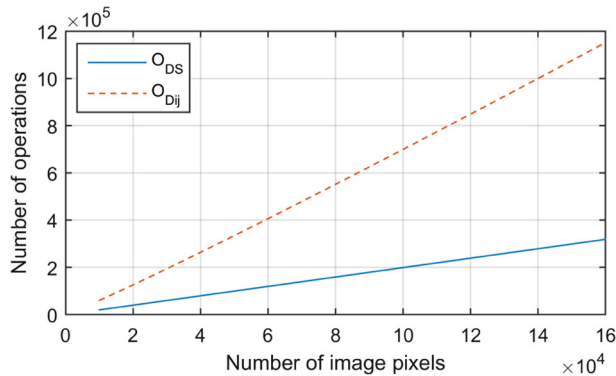
In each step of the double scan algorithm, we perform the following calculations. First, we compute the absolute intensity difference between the values from PSI array  $S$  for pixels  $P_2$ ,  $P_3$  and the grayscale value of pixel  $P_1$ , and thus we create two initial isoline distance estimations for pixel  $P_1$ :

$$\mathcal{J}'_2 = |S_2 - I_1|, \quad \mathcal{J}'_3 = |S_3 - I_1|, \quad (5)$$

where  $\mathcal{J}'_2$  and  $\mathcal{J}'_3$  are the preliminary isoline distance estimates,  $S_2$  and  $S_3$  are the values of the PSI array for pixels  $P_2$  and  $P_3$ ,  $I_1$  is the grayscale intensity of pixel  $P_1$ .

In the next step, we compare the two above estimates with the isoline distance of the corresponding adjacent pixels and we always choose the greater one, so that our isoline distance will never decrease, (consider Figs. 2 and 3):

$$\text{if } (\mathcal{J}'_k < \mathcal{J}_k) \text{ then } (\mathcal{J}'_k \leftarrow \mathcal{J}_k), \quad (6)$$



**Fig. 6** Comparison of computational burden of Dijkstra's algorithm (red line) and double-scan algorithm (blue line)

where  $\mathcal{J}_k$  is the isoline distances of pixel  $P_k$ , ( $k = 2, 3$ ).

Finally, in the last part of our algorithm, pixel  $P_1$  is given the smallest isoline distance from the following values:  $\mathcal{J}'_2$ ,  $\mathcal{J}'_3$  and  $\mathcal{J}_1$ :

$$\mathcal{J}_1 \leftarrow \min(\mathcal{J}'_2, \mathcal{J}'_3, \mathcal{J}_1). \quad (7)$$

Additionally, since each of the three values has its own assigned seed intensity, we transfer the selected one to pixel  $P_1$ , according to the smallest value selected:

$$\begin{aligned} &\text{if (smallest} = \mathcal{J}'_2) \text{ then } (S_1 \leftarrow S_2), \\ &\text{if (smallest} = \mathcal{J}'_3) \text{ then } (S_1 \leftarrow S_3), \\ &\text{if (smallest} = \mathcal{J}_1) \text{ then } (S_1 \leftarrow S_1). \end{aligned} \quad (8)$$

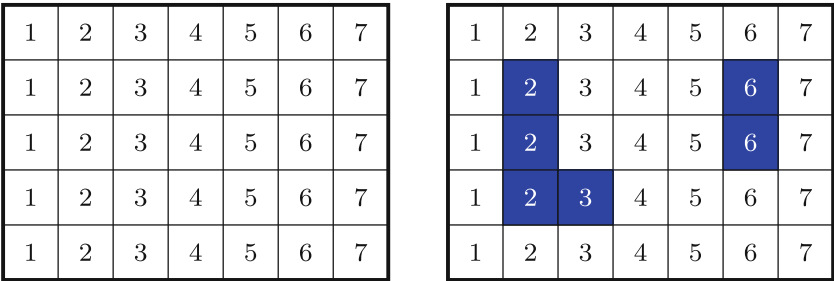
If the isoline distance has already been calculated in the previous scan, then it is possible, that initially the estimated distances will be larger in the second scan because of the presence of two (or more) scribbles of the same colour. In this case, the scribbles (the seed pixels) seem to compete with each other and the one, whose path to a given pixel has lower total intensity deviations, wins.

After performing the running window double scan operations, each pixel of the image receives its isoline distance. As can be observed, only four corner pixels are excluded from the analysis. Their values can easily be replaced using one of the adjacent pixels. In our algorithm, we decided to replace it with the mean of all three neighbours. Figure 7 illustrates an example of the results of the first and second scan for both the propagated seed intensity and the isoline distance arrays. We also provide a pseudo code to calculate the isoline distance (Algorithm 1).

When the isoline distance maps are completed, the colour of each pixel can finally be determined using chrominance weighted averaging performed in the YCrCb colour space. The luminance Y component is obtained directly from the grayscale image while the new chrominance values are assigned to pixels using the weighted average of the chrominance values of inserted scribbles:

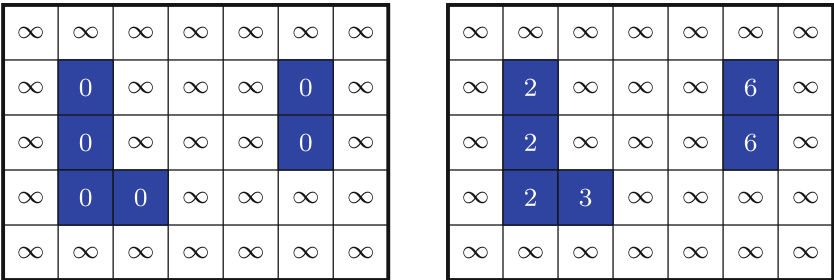
$$\mathcal{C}_j = \frac{\sum_{i=1}^N C_i \exp\{-\frac{\mathcal{J}_{i,j}}{\sigma_i}\}}{\sum_{i=1}^N \exp\{-\frac{\mathcal{J}_{i,j}}{\sigma_i}\}}, \quad (9)$$





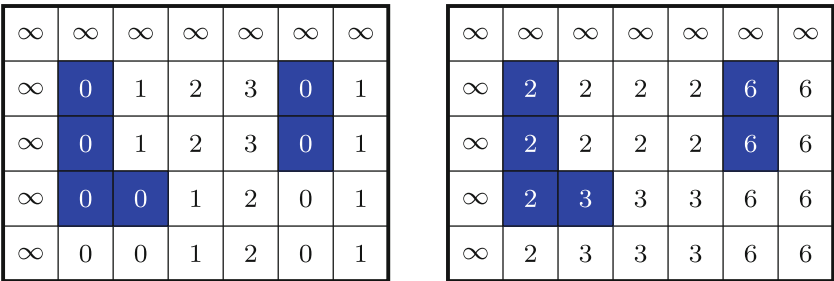
a) Test image - the values represent the intensities of the pixels.

b) Test image with the scribbles.



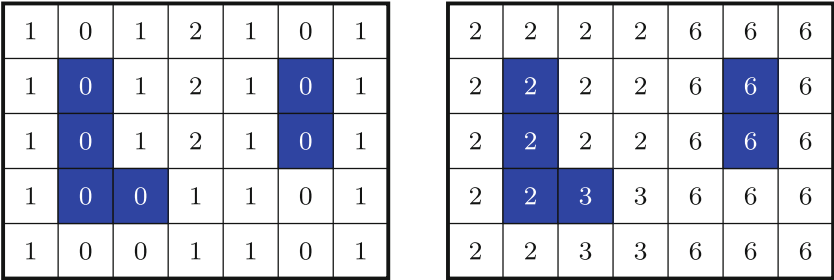
(c) Initialisation of the isoline distance array.

(d) Initialisation of PSI array  $S$ .



(e) Distance array after the 1 scan.

(f) PSI array  $S$  after the 1 scan.



(g) Distance array after the 2 scan.

(h) PSI array  $S$  after the 2 scan.

**Fig. 7** Example of the calculation of the isoline distance and seed intensity arrays

**Algorithm 1** Pseudocode for isoline distance calculation.**Definitions:**

$M, N$  - width and height of image  
 $R$  - number of double-scan repetitions  
 $I(x, y)$  - intensity of pixel at  $(x, y)$   
 $\mathcal{J}(x, y)$  - isoline distance at  $(x, y)$   
 $S(x, y)$  - propagated seed intensity at  $(x, y)$   
 $\Omega$  - set of scribbled pixels positions

**Initialisation:**

```

if  $(x, y) \in \Omega$  then
  |  $\mathcal{J}(x, y) = 0$ ;  $S(x, y) = I(x, y)$ ;
else
  |  $\mathcal{J}(x, y) = \infty$ ;  $S(x, y) = \infty$ ;
end
  
```

```

for  $repetition = 1 : R$  do
  | First scan (left to right, top to bottom):
  | for  $y = 2 : N$  do
  | | for  $x = 2 : M$  do
  | | |  $\mathcal{J}'_1 = |S(x, y - 1) - I(x, y)|$  ;
  | | |  $\mathcal{J}'_2 = |S(x - 1, y) - I(x, y)|$  ;
  | | | if  $\mathcal{J}'_1 < \mathcal{J}(x, y - 1)$  then
  | | | |  $\mathcal{J}'_1 = \mathcal{J}(x, y - 1)$ ;
  | | | end
  | | | if  $\mathcal{J}'_2 < \mathcal{J}(x - 1, y)$  then
  | | | |  $\mathcal{J}'_2 = \mathcal{J}(x - 1, y)$ ;
  | | | end
  | | |  $\mathcal{J}(x, y) = \min\{\mathcal{J}'_1, \mathcal{J}'_2, \mathcal{J}(x, y)\}$ ;
  | | | switch  $\mathcal{J}(x, y)$  do
  | | | | case  $\mathcal{J}'_1$ 
  | | | | |  $S(x, y) = S(x, y - 1)$ ;
  | | | | end
  | | | | case  $\mathcal{J}'_2$ 
  | | | | |  $S(x, y) = S(x - 1, y)$ ;
  | | | | end
  | | | endsw
  | | endsw
  | end
  | end
  | Second scan (left to right, top to bottom):
  | for  $y = N - 1 : 1$  do
  | | for  $x = M - 1 : 1$  do
  | | |  $\mathcal{J}'_1 = |S(x, y + 1) - I(x, y)|$  ;
  | | |  $\mathcal{J}'_2 = |S(x + 1, y) - I(x, y)|$  ;
  | | | if  $\mathcal{J}'_1 < \mathcal{J}(x, y + 1)$  then
  | | | |  $\mathcal{J}'_1 = \mathcal{J}(x, y + 1)$ ;
  | | | end
  | | | if  $\mathcal{J}'_2 < \mathcal{J}(x + 1, y)$  then
  | | | |  $\mathcal{J}'_2 = \mathcal{J}(x + 1, y)$ ;
  | | | end
  | | |  $\mathcal{J}(x, y) = \min\{\mathcal{J}'_1, \mathcal{J}'_2, \mathcal{J}(x, y)\}$ ;
  | | | switch  $\mathcal{J}(x, y)$  do
  | | | | case  $\mathcal{J}'_1$ 
  | | | | |  $S(x, y) = S(x, y + 1)$ ;
  | | | | end
  | | | | case  $\mathcal{J}'_2$ 
  | | | | |  $S(x, y) = S(x + 1, y)$ ;
  | | | | end
  | | | endsw
  | | endsw
  | end
  | end
end
  
```

where  $C_j$  is an assigned chrominance (Cr or Cb) to the pixel at position  $j$ ,  $C_i$  is the corresponding chrominance of the  $i$ -th scribble,  $\mathcal{J}_{i,j}$  is the calculated isoline distance from the  $i$ -th scribble to the pixel at position  $j$ ,  $\sigma_i$  is the parameter that is responsible for the level of colour blending assigned to the  $i$ -th scribble. In the experiments we did not discriminate the scribbles, so that  $\sigma_i = \sigma$ , where  $\sigma$  is an experimentally found, default parameter value.

### 3.2 Algorithm parameters

In order to allow the results of our colourisation algorithm to be modified, we introduced two additional parameters that can be adjusted to obtain satisfactory colourisation effects. Their impact on the performance of the algorithm is discussed in detail in Section 4. These parameters allow the sensitivity of the colourisation to be increased to the topographic distance on the image domain or to the intensity deviation between the image pixels and the seeds of the scribbles.

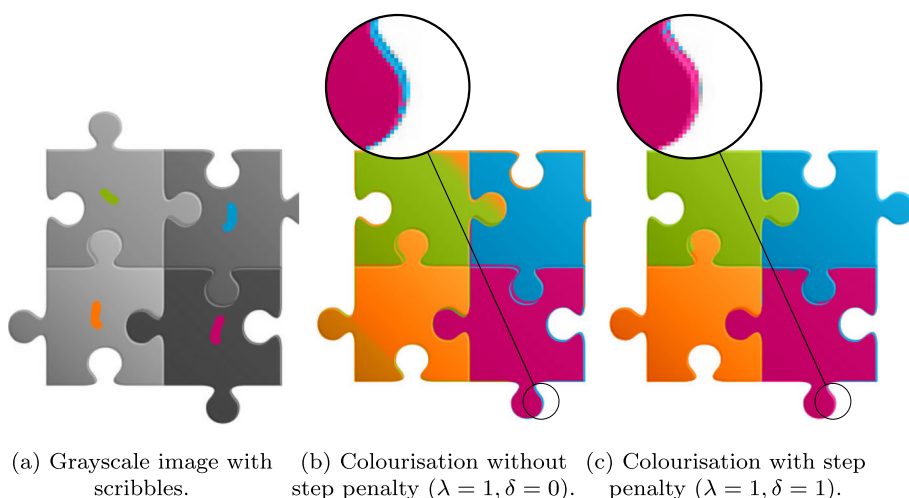
The first parameter  $\lambda$  is used to raise the absolute difference between adjacent pixel intensities to the power and the second parameter  $\delta$  is the step penalty that is added to each transition between neighbouring pixels. These parameters lead to the following modification of (5):

$$\mathcal{J}'_2 = |S_2 - I_1|^\lambda + \delta, \quad \mathcal{J}'_3 = |S_3 - I_1|^\lambda + \delta. \quad (10)$$

Increasing the  $\lambda$  parameter is primarily useful for smooth images and the introduction of  $\delta$  is beneficial as it prevents the so-called colour leakage that is sometimes visible in image regions that are far from a given scribble. The improvement of colourisation after applying the step penalty  $\delta$  can be observed in Fig. 8.

### 3.3 Improvement in computational speed

We noticed (as in [21]), that for most of the images, there is usually no need to mix more than three colours in order to achieve satisfactory results. Therefore, only the most significant colours, i.e. the closest to a given pixel, were considered in the colour blending phase



**Fig. 8** Colour leakage reduction using the step penalty  $\delta$

defined by (9). Thus, the calculation of the isoline distance from a given scribble to each pixel (especially the one very far from the scribble) is not the optimal solution. To remove these redundant calculations we propose the following modification of our algorithm.

Instead of creating distance maps for each colour, we operate on three such maps - for the most prominent colour scribbles. Similarly, we use three chrominance arrays that hold the colour chrominances of the three closest scribbles. The seed arrays are also tripled. By doing this, the computation time and the memory usage are independent of the number of scribbles.

The first step of the algorithm, previously described by (5), is modified so that we can calculate the isoline distance from the two adjacent pixels and from each of their three closest scribbles  $\gamma_1$ ,  $\gamma_2$  and  $\gamma_3$ :

$$\mathcal{J}'_{2|\gamma_k} = |S_{2|\gamma_k} - I_1|^\lambda + \delta, \quad \mathcal{J}'_{3|\gamma_k} = |S_{3|\gamma_k} - I_1|^\lambda + \delta, \quad (11)$$

where  $\mathcal{J}'_{2,\gamma_k}$  and  $\mathcal{J}'_{3,\gamma_k}$  are the isoline distance estimations from the  $k$ -th colour scribble,  $S_{2|\gamma_k}$  and  $S_{3|\gamma_k}$  are the PSI values of pixels  $P_2$  and  $P_3$  from the seed array of the  $k$ -th scribble, ( $k = 1, 2, 3$ ). This permits the total number of the six isoline distance estimates to be obtained rather than the two in the original algorithm. Moreover, we do not have to repeat the calculations for each scribble independently.

In the next step, we compare the isoline distances of adjacent pixels with the estimates described above, and apply the same conditional correction as in (6):

$$\text{if } \mathcal{J}'_{k|\gamma_q} < \mathcal{J}_{k|\gamma_q} \text{ then } \mathcal{J}'_{k|\gamma_q} \leftarrow \mathcal{J}_{k|\gamma_q}, \quad (12)$$

where  $k = 2, 3$  and  $q = 1, 2, 3$ .

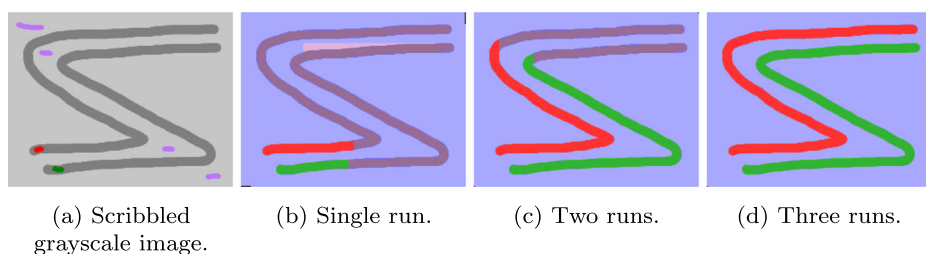
At the end, we create an array which consists of six estimated distances and three own distances of the pixel  $P_1$ . The three smallest isoline distances are selected from this collection and their corresponding seed intensities and chrominances are transferred to pixel  $P_1$ .

In the final colour weighted averaging, there are always three colour candidates whose distance and chrominance can be obtained directly from the three arrays. Although this modification can easily be extended to a larger number of colours, we did not observe a significant improvement in the quality of the colourisation. It is sometimes actually better to reduce the number of colours to two. Very often, this is enough to achieve satisfactory colourisation results and such a simplification leads to a further improvement in the speed of the algorithm.

### 3.4 Double scan repetition

The double-scan algorithm is a very fast and a simple method to estimate distance in our application. However, it has one disadvantage. Due to the directions of the scan, it is impossible to accurately estimate the distance of sharply curved paths. This leads to considerable problems in the colourisation of more complicated shapes.

Repetition of the double scan can resolve this issue. We obtained very good results for all the colourised images after three runs of the double scan experimentally, and therefore it was selected as the default number of repetitions in all of the experiments (increasing the number of repetitions did not result in any further improvement in quality). However, we would like to mention, that this number may depend on the complexity of the image and the positioning of the scribbles. In order to use colourisation for the segmentation of curved objects, the required number of repetitions may vary and therefore it should be chosen experimentally (see the exemplary test in Fig. 9).



**Fig. 9** Repetition of the double-scan algorithm scanning

## 4 Parameter optimisation

In the proposed method, a user can change three parameters. Parameter  $\lambda$  tunes the impact of any intensity changes on the colourisation results,  $\delta$  is the step penalty, that reduces the colour leakage and  $\sigma$  is responsible for colour blending. We observed that for most grayscale images with intensities in the range of  $[0, 255]$ , the last parameter can be set at  $\sigma = 10$  without a drop of quality. This means that the operator only has to adjust two parameters.

Comparison of the colourisation results can be difficult because the reference colour image is usually unknown. However, to present the influence of the parameters on the colourisation quality, we converted the original colour images to a grayscale version by only taking the Y luminance channel, and then we colourised them with our algorithm. Because the reference is known, this procedure permits the utilisation of the colour image quality measures.

We applied the Peak Signal-to-Noise Ratio (PSNR) as it is one of the most widely used methods for image comparison. For colour images and 8-bit colour representation, PSNR is calculated using the mean square error (MSE) as follows:

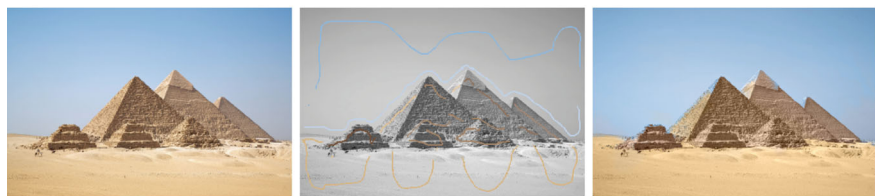
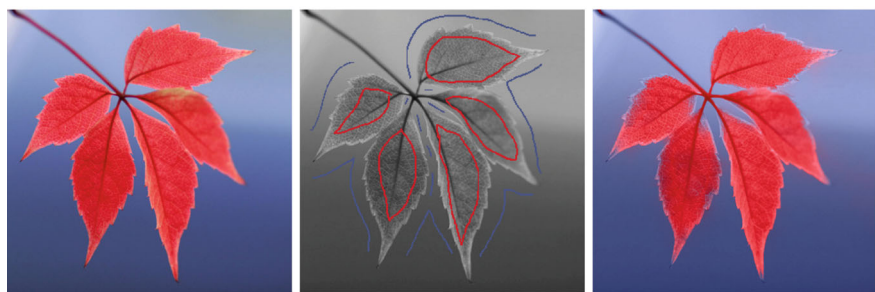
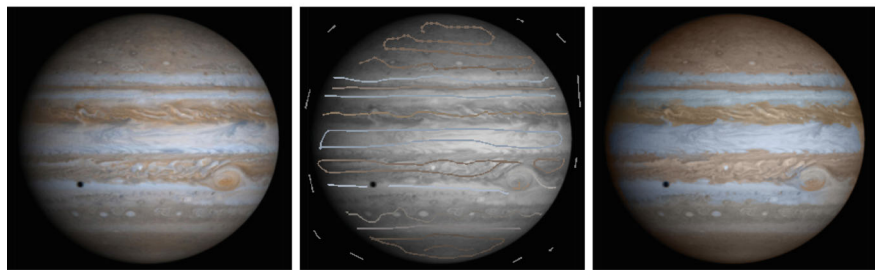
$$\text{MSE} = \frac{1}{3NM} \sum_{q \in \{R, G, B\}} \sum_{i=1}^N \sum_{j=1}^M [I^q(i, j) - I_C^q(i, j)]^2,$$

$$\text{PSNR} = 10 \log \left( \frac{255^2}{\text{MSE}} \right), \quad (13)$$

where  $N$ ,  $M$  are width and height of the image,  $q$  is the colour channel (R, G or B),  $I^q(i, j)$  and  $I_C^q(i, j)$  are the intensities of the pixel at position  $(i, j)$  in the reference and colourised image, respectively.

We chose the Normalised Colour Difference (NCD) [12, 17] operating in the  $L^*a^*b^*$  colour space as the second indicator of the quality of the colourisation. This coefficient represents the distances between colours in a given colour space. Since the larger the NCD is, the worse the image quality is, we decided to visualise  $1/\text{NCD}$  instead of NCD in order to improve the readability of presented optimisation plots. The NCD indicator is calculated using the following formula:

$$\text{NCD} = \frac{\sum_{i=1}^N \sum_{j=1}^M \sqrt{\sum_{q \in \{L^*, a^*, b^*\}} [I^q(i, j) - I_C^q(i, j)]^2}}{\sum_{i=1}^N \sum_{j=1}^M \sqrt{\sum_{q \in \{L^*, a^*, b^*\}} I^q(i, j)^2}}. \quad (14)$$



Original colour image.

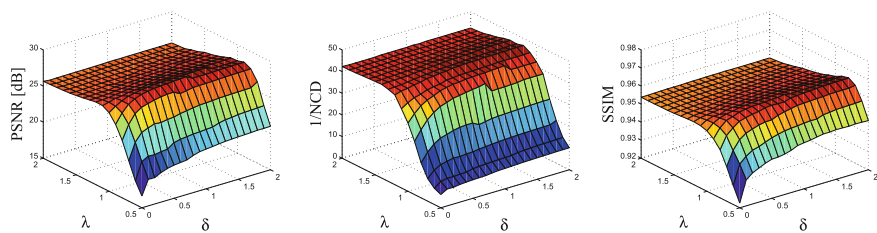


Scribbled image.

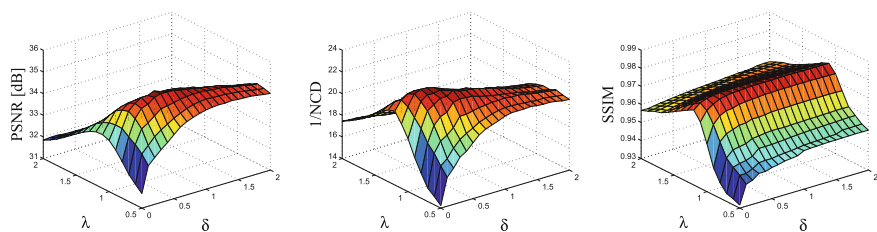


Colourised image.

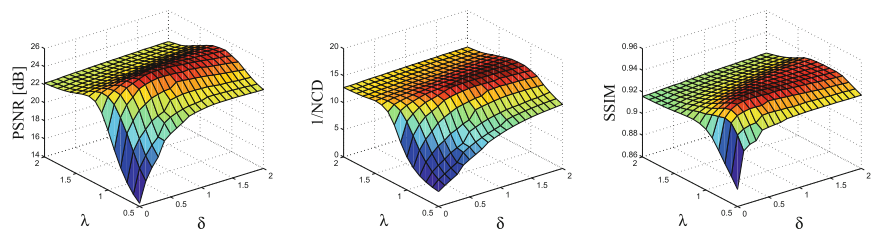
**Fig. 10** The test images used for the parameter optimisation and for the comparison with the state-of-the-art colourisation methods. From top: Balloons, Jupiter, Leaf, Pyramids and Puzzle



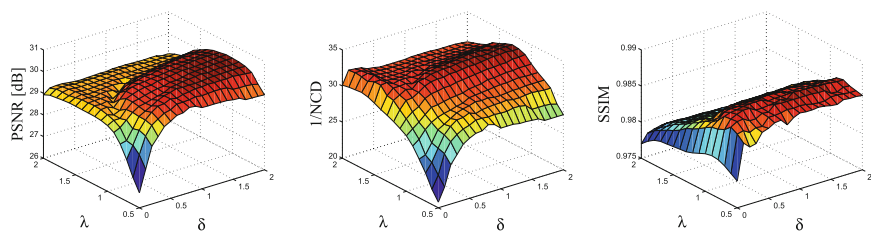
Test image BALLOONS.



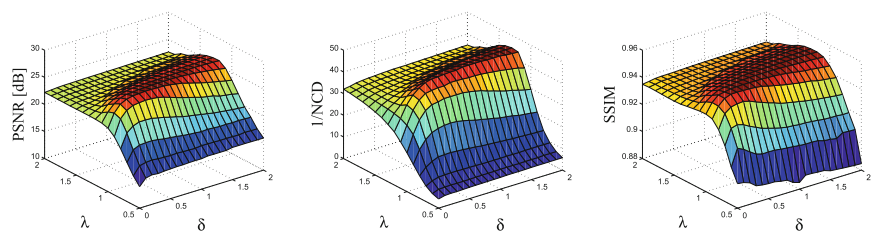
Test image JUPITER.



Test image LEAF.



Test image PYRAMIDS.



Test image PUZZLE.

**Fig. 11** Dependence of applied quality measures on the parameters  $\lambda$  and  $\delta$

The third quality measure is the Structural Similarity Index (SSIM) [18]. This coefficient is less concentrated on the intensity difference than the both of the previous ones. Rather, its results correspond very well with the human perception. Because the SSIM is only calculated in the local window (in our tests  $5 \times 5$  pixels), we obtained the SSIM as the mean of SSIM values over the entire image. The SSIM can be calculated for a single colour channel, and therefore we averaged the results for each (R, G, B) in order to obtain the final similarity estimate of the colour image. For detailed formulas for calculating the structural similarity index, refer to [18].

In our studies we used five colour test images that represented different synthetic and real image types (see the first column in Fig. 10). They were scribbled and colourised using slightly different combinations of the parameters. The 3-D plots (see Fig. 11) present the results of the colourisation quality using the corresponding quality measures. We also summarise the optimal  $\lambda$  and  $\delta$  settings in Table 1.

We observed that the optimal results were obtained with a combination of  $\lambda - \delta$  parameters in the following range:  $\lambda \in (0.5, 2)$  and  $\delta \in (0, 2)$ . We should also note that the surface plots for each parameter had very similar shapes, which indicate that the above range of the  $\lambda - \delta$  combinations produces very similar and satisfactory quality of colourisation.

We performed test with more images and assessed the colourisation quality visually. We found that the most pleasing results were obtained for the combination of  $\lambda = 1$  and  $\delta = 1$  which fits well the previously specified ranges. Therefore, this set of parameters values should be used as a default pair.

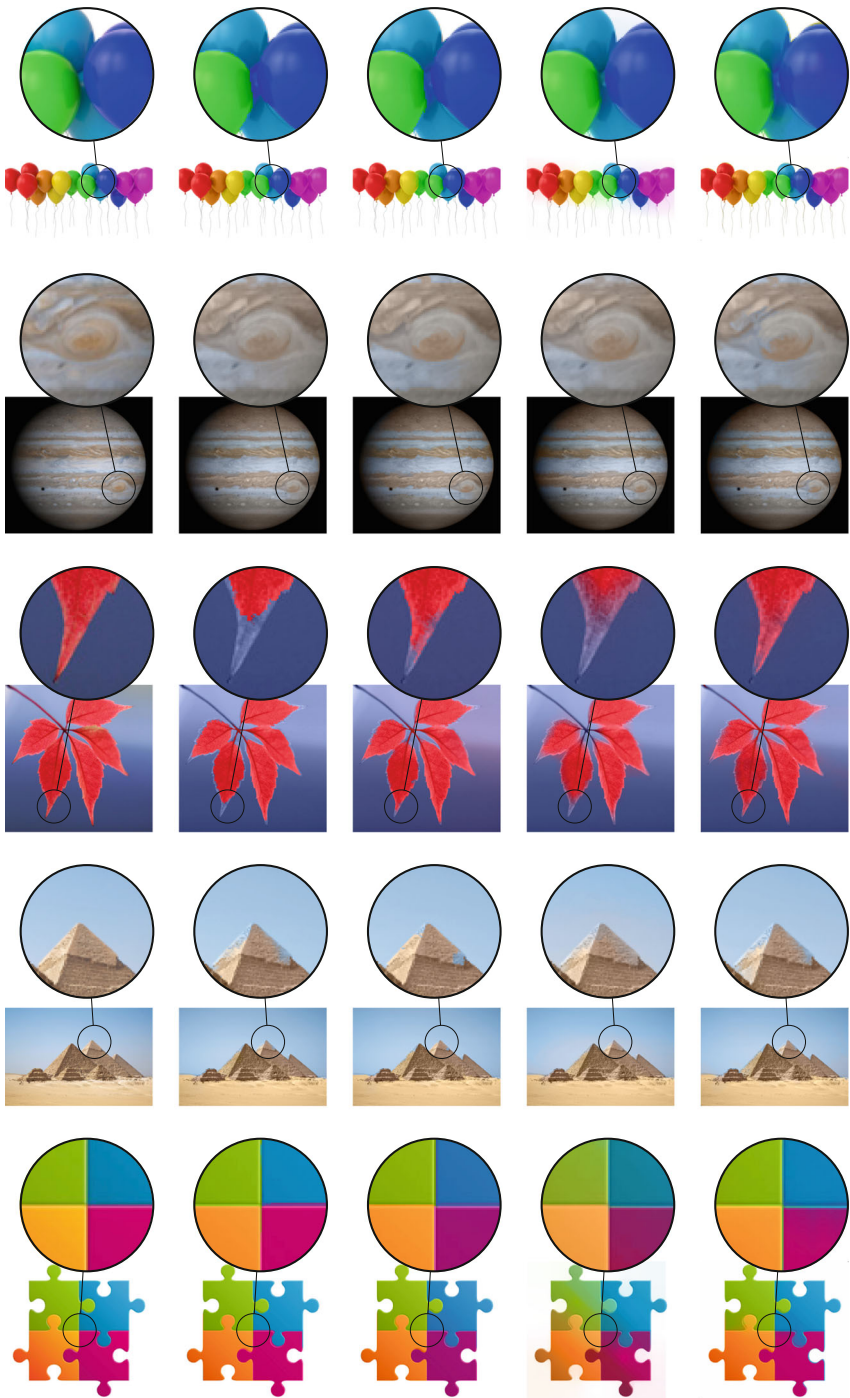
## 5 Comparison of the colourisation methods

We compared the results of the colourisation of the grayscale versions of the test images with three well established methods. We selected the ones, that are frequently cited and which, like to the presented algorithm, utilise the interactive colourisation that is based on user-indicated scribbles. For the first one - *Colourisation Using Optimisation* (CUO) [7] - we used the Matlab code that is available on-line. The two remaining algorithms were implemented in C# language: *Chrominance Blending Colourisation* utilizing Dijkstra's algorithm (CBC) [21] and the *Distance Transform based Colourisation* (DTC) [6]. As no code is available for method [21], it was implemented in its basic version. For our method we used ISO abbreviation.

**Table 1** Optimal parameter selection

QM	Balloons	Jupiter	Leaf	Pyramid	Puzzle
PSNR	28.45	35.50	25.58	30.39	26.45
	$\delta=1.1$	$\delta=0.6$	$\delta=0.8$	$\delta=2.0$	$\delta=0.6$
	$\lambda=0.9$	$\lambda=0.8$	$\lambda=1.0$	$\lambda=1.0$	$\lambda=1.0$
1/NCD	45.87	22.57	15.63	33.56	44.84
	$\delta=0.9$	$\delta=0.4$	$\delta=1.1$	$\delta=1.7$	$\delta=1.1$
	$\lambda=1.2$	$\lambda=1.1$	$\lambda=1.1$	$\lambda=1.3$	$\lambda=1.2$
SSIM	0.965	0.978	0.942	0.986	0.951
	$\delta=2.0$	$\delta=1.3$	$\delta=0.5$	$\delta=1.2$	$\delta=0.6$
	$\lambda=0.8$	$\lambda=1.1$	$\lambda=0.8$	$\lambda=0.8$	$\lambda=1.0$





Original image. CBC [21]. DTC [6]. CUO [7]. Proposed.

**Fig. 12** Results of colourisation using different algorithms

**Table 2** Comparison of colourisation quality

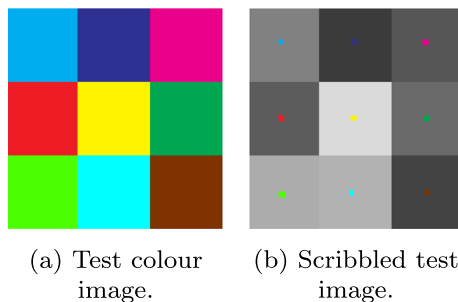
	QM	CBC	CUO	DTC	ISO
Balloons	PSNR	28.1	27.49	28.13	28.45
	1/NCD	49.26	31.94	48.54	41.66
	SSIM	0.9609	0.9632	0.9611	0.9639
JUPITER	PSNR	34.47	32.92	33.99	35.5
	1/NCD	22.2	21.32	18.83	22.12
	SSIM	0.9759	0.9686	0.979	0.9582
Leaf	PSNR	23.31	24.55	24.66	25.58
	1/NCD	14.06	14.08	13.44	14.97
	SSIM	0.9326	0.9412	0.9371	0.9413
Pyramids	PSNR	29.58	31.54	29.26	30.39
	1/NCD	28.9	35.33	28.16	32.36
	SSIM	0.9866	0.9861	0.9863	0.9851
Puzzle	PSNR	27.89	21.38	25.09	26.45
	1/NCD	49.01	12.83	23.41	37.73
	SSIM	0.9539	0.9165	0.9354	0.9459

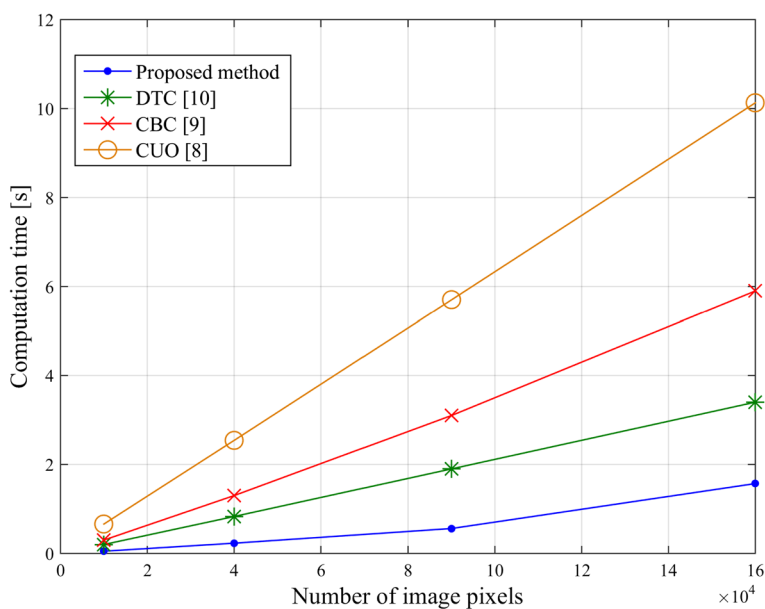
Best results are presented in bold type

The grayscale versions of the test images were colourised using identical scribble arrangements as depicted in the middle column of Fig. 10. For our method and each test image we used the parameter combination that provided the highest PSNR (see Table 1). Figure 12 shows the outcomes of each method. We compared the quality of the colourisation using the quality metrics (QM) described previously. Table 2 presents the summary.

The comparison that was performed proved the high quality of the isoline based colourisation results. The images that were colourised using the novel algorithm are of a similar, or sometimes even better, quality when compared with the other algorithms. Moreover, a visual examination confirmed that the proposed algorithm yields visually pleasing results.

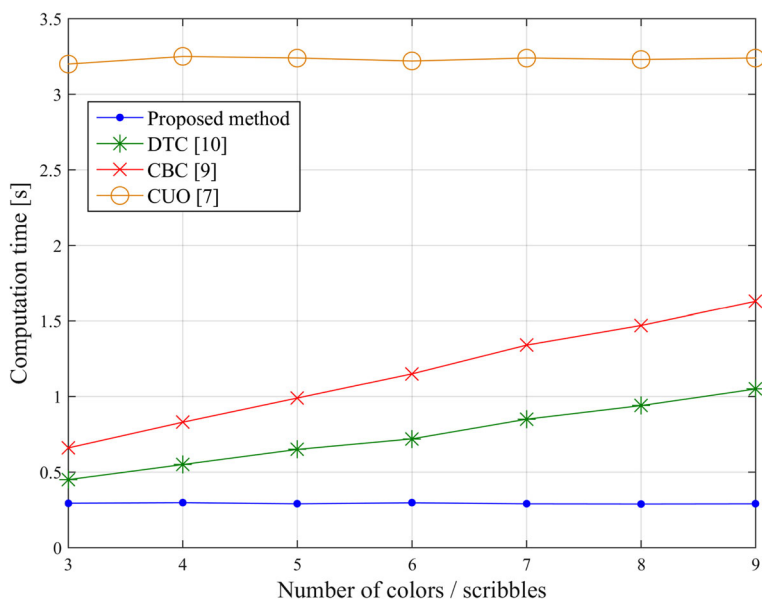
However, it must be clearly stated that the quality of the colourisation of numerous methods (not only the ones selected in this evaluation) is very high and the results are visually comparable. Moreover, the scribble configuration can, to some extent, influence the final results, thus changing the “winners” in each of the images tested. Therefore, we would like to note that the main goal of our comparison was not to prove the superiority of the outputs obtained by the proposed method. Rather, it was conducted to illustrate that the quality of the colourisation quality is at the same, satisfactory level.

**Fig. 13** Chessboard image used for computational speed tests

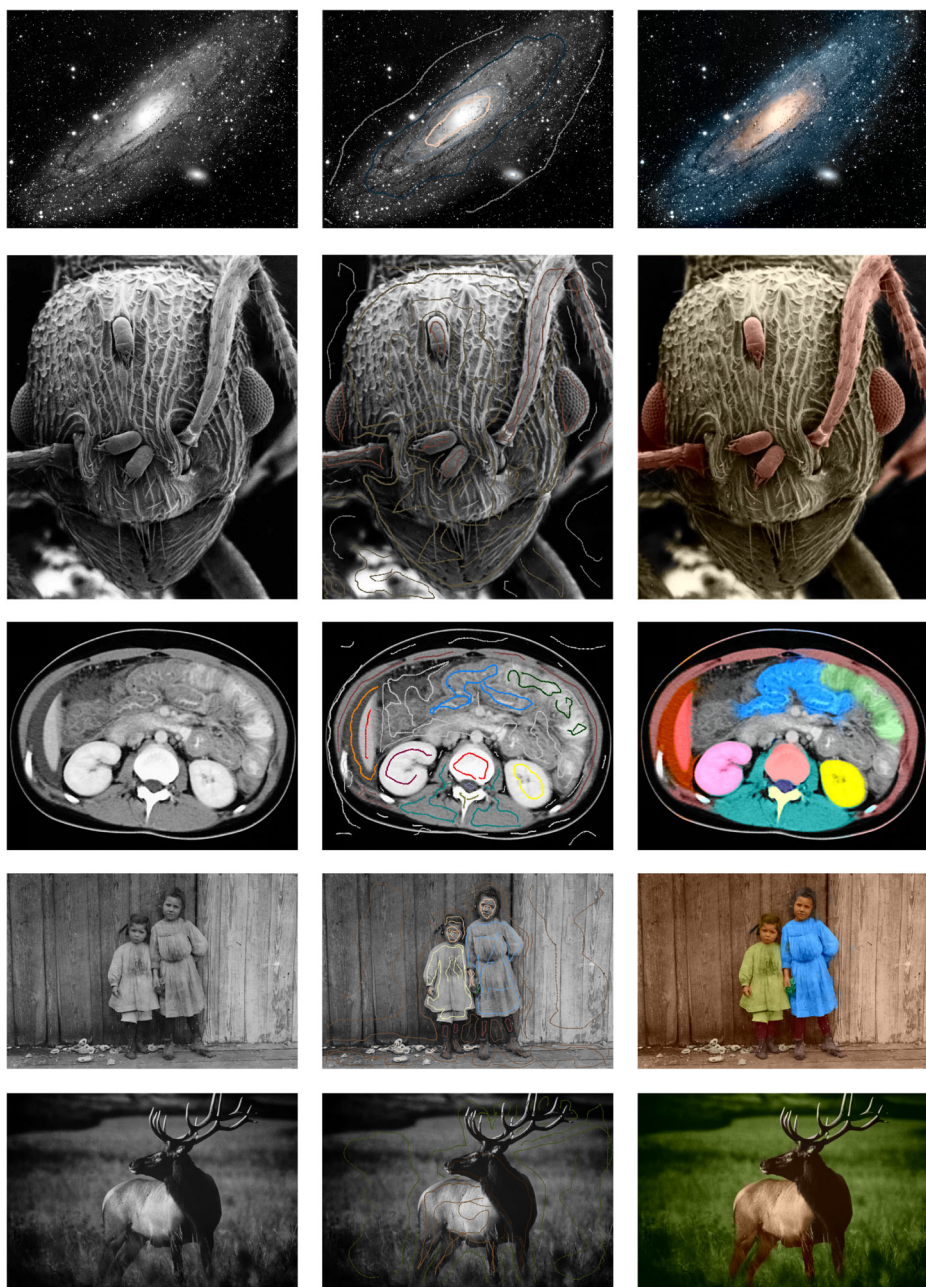


**Fig. 14** Dependence of the computation time on the number of image pixels

Because the fine details of the final result depend on the configuration of the scribbles and because the values of the parameters can be changed interactively by a user, the computational efficiency is of the highest priority. Furthermore, since the quality of the coloured images obtained using various methods is comparable, the algorithm speed should be treated as the most significant factor that has a direct influence on the time a user spends colourising.



**Fig. 15** Dependence of the computation time on the number of colour scribbles



Original image.

Scribbled image.

Colourisation result.

**Fig. 16** Exemplary application of the proposed isoline colourisation technique

In order to assess the computational burden of the described technique we used the simple test image depicted in Fig. 13. By changing the number of scribbles (three to nine scribbles) and the resolution of the grayscale image, we obtained the computational characteristics of

the algorithms. The colourisation application with the competitive methods implemented worked in the Windows<sup>7</sup> environment (Intel Core i5-3470 CPU@3.2Ghz, 16GB RAM DDR3) and its process was set to the maximum priority. We turned off all unnecessary applications, thus the system background could not significantly affect the measurement of the computational time. Figures 14 and 15 present the corresponding results.

The tests proved the high computational efficiency of our framework. Isoline colourisation provided the lowest calculation times for every combination of the number of scribbles and image size. We would like to mention that all of the methods showed a linear dependency on the number of image pixels. This is consistent with the theoretical predictions presented in Section 3. The predominance of double-scan (ISO) over Dijkstra's algorithm (CBC) was visible as was expected (see Fig. 6). The DTC algorithm also utilises the double-scan method, and therefore it is faster than CBC. However, in this colourisation, the double-scan was repeated for each colour scribble, while in our approach, it was limited to the three closest colours, which caused the difference in efficiency. The CUO algorithm utilises completely different colourisation approach (i.e. not based on distance maps) which appears to be significantly slower than the others.

The situation is different for the impact of the number of scribbles. Our isoline-based approach and CUO did not exhibit an increase in computational time with a growing number of scribbles. This is a very important advantage when colourising complex images, that require a greater number of scribbled details.

## 6 Applications

Colourisation can be applied in medicine, astronomy, biology and other disciplines. We should also consider colourisation for educational purposes as colour images are more attractive to a viewer who can benefit from such visualisation. The indication of some crucial regions within medical images may be of special interest to the physicians. Astrophotography can also take advantage of colourisation when capturing of colour images requires separate images obtained through RGB filters. With proper colourisation, such a procedure may be reduced to only single grayscale image photography. Figure 16 presents a few examples of the colourisation that was obtained using our algorithm.

## 7 Conclusion

In this work we presented a novel, very fast algorithm for image colourisation that is based on the introduced isoline concept. In our interactive colourisation method, a user indicates the colour scribbles in order to obtain the required colourisation result. By using a repeated double-scan algorithm, reducing the number of mixed colours and also by enabling parameter tuning, our method obtained a high accuracy and computational efficiency.

The method was compared with three well-known colourisation algorithms. The comparison included the aspects of computational complexity and the quality of the colourisation. We proved that our method can achieve satisfactory quality measured by objective fidelity metrics. The assessment of the results confirms that the outputs are comparable to, or sometimes better than, the results obtained using the competitive algorithms. Our method also showed a distinctively higher computational efficiency, which does not depend on the number of indicated scribbles. This is an important feature, which combined with the simplicity of its implementation, makes the proposed colourisation method very attractive.



**Acknowledgments** The research leading to these results received funding from the Project Pol-Nor/204256/16/2013 carried out within the Polish-Norwegian Research Programme, Norwegian Financial Mechanism 2009–2014 and was performed using the infrastructure supported by POIG.02.03.01-24-099/13 grant: GeCONiI - Upper Silesian Center for Computational Science and Engineering.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

1. Horiuchi T, Hirano S (2003) Colorization algorithm for grayscale image by propagating seed pixels. In: Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on, vol. 1, pp. 1–457–60 vol. 1. doi:[10.1109/ICIP.2003.1246997](https://doi.org/10.1109/ICIP.2003.1246997)
2. Jacob V, Gupta S (2009) Colorization of grayscale images and videos using a semiautomatic approach. In: Image Processing (ICIP), 2009 16th IEEE International Conference on, pp 1653–1656. doi:[10.1109/ICIP.2009.5413392](https://doi.org/10.1109/ICIP.2009.5413392)
3. Kawulok M, Kawulok J, Smolka B (1722) Discriminative textural features for Image and video colorization. IEICE Trans Inf Syst 95:1722–1730. doi:[10.1587/transinf.E95.D](https://doi.org/10.1587/transinf.E95.D)
4. Kawulok M, Smolka B (2010) Competitive image colorization. In: Image Processing (ICIP), 2010 17th IEEE International Conference on, pp 405–408. doi:[10.1109/ICIP.2010.5653544](https://doi.org/10.1109/ICIP.2010.5653544)
5. Kawulok M, Smolka B (2011) Texture-adaptive image colorization framework. EURASIP J Adv Sig Proc 2011:99
6. Lagodzinski P, Smolka B (2014) Application of the extended distance transformation in digital image colorization. Multimedia Tools Appl 69(1):111–137
7. Levin A, Lischinski D, Weiss Y (2004) Colorization using optimization. ACM Trans Graph 23(3):689–694. doi:[10.1145/1015706.1015780](https://doi.org/10.1145/1015706.1015780)
8. Levin A, Lischinski D, Weiss Y (2004) Colorization using optimization. In: ACM SIGGRAPH 2004 Papers, SIGGRAPH '04, pp. 689–694. ACM, New York, NY, USA. doi:[10.1145/1186562.1015780](https://doi.org/10.1145/1186562.1015780)
9. Markle W (1984) The development and application of colorization. Soc Motion Picture Television Eng 93(7):632–635. doi:[10.5594/J03526](https://doi.org/10.5594/J03526)
10. Mehlhornt K, Sanders P (2008) Algorithms and data structures. Springer, Berlin
11. Osma-Ruiz V, Godino-Llorente JI, Sáenz-Lechón N., Gómez-Vilda P. (2007) An improved watershed algorithm based on efficient computation of shortest paths. Pattern Recogn 40(3):1078–1090. doi:[10.1016/j.patcog.2006.06.025](https://doi.org/10.1016/j.patcog.2006.06.025)
12. Plataniotis K, Venetsanopoulos A (2000) Color image processing and applications, Springer, Berlin
13. Reinhard E, Adhikhmin M, Gooch B, Shirley P (2001) Color transfer between images. IEEE Comput Graph Appl 21(5):34–41
14. Rosenfeld A, Pfaltz J (1968) Distance functions on digital pictures. Pattern Recogn 1(1):33–61. doi:[10.1016/0031-3203\(68\)90013-7](https://doi.org/10.1016/0031-3203(68)90013-7)
15. Rosenfeld A, Pfaltz J (1968) Distance functions on digital pictures. Pattern Recogn 1(1):33–61
16. Ruderman DL, Cronin TW, Chiao CC (1998) Statistics of cone responses to natural images: Implications for visual coding. J Opt Soc Am A 15:2036–2045
17. Russo F (2014) Performance evaluation of noise reduction filters for color images through norMalized color difference (NCD) decomposition ISRN Machine Vision 579658
18. Wang Z, Bovik A, Sheikh H, Simoncelli E (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612. doi:[10.1109/TIP.2003.819861](https://doi.org/10.1109/TIP.2003.819861)
19. Welsh T, Ashikhmin M, Mueller K (2002) Transferring color to greyscale images. ACM Trans Graph 21(3):277–280. doi:[10.1145/566654.566576](https://doi.org/10.1145/566654.566576)
20. Yan WQ, Kankanhalli M (2003) Colorizing infrared home videos. In: Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on, vol. 1, pp. 1–97–100 vol.1. doi:[10.1109/ICME.2003.1220863](https://doi.org/10.1109/ICME.2003.1220863)
21. Yatziv L, Sapiro G (2006) Fast image and video colorization using chrominance blending. IEEE Trans Image Process 15(5):1120–1129. doi:[10.1109/TIP.2005.864231](https://doi.org/10.1109/TIP.2005.864231)
22. Zhang Z, Cui H, Lu H, Chen R, Yan Y (2009) A colorization method based on fuzzy clustering and distance transformation. In: Image and Signal Processing, 2009. CISP '09. 2nd International Congress on, pp 1–5. doi:[10.1109/CISP.2009.5304487](https://doi.org/10.1109/CISP.2009.5304487)

23. Zhao Y, Wang L, Jin W, Shi S (2007) Colorizing biomedical images based on color transfer. In: Complex Medical Engineering, 2007. CME 2007. IEEE/ICME International Conference on, pp 820–823. doi:[10.1109/ICCME.2007.4381855](https://doi.org/10.1109/ICCME.2007.4381855)
24. Zhen Z, Yan G, Lizhuang M (2012) An automatic image and video colorization algorithm based on pattern continuity. In: Audio, Language and Image Processing (ICALIP), 2012 International Conference on, pp 531–536. doi:[10.1109/ICALIP.2012.6376674](https://doi.org/10.1109/ICALIP.2012.6376674)



**Adam Popowicz** received M. Sc. Degree in electronics from the Silesian University of Technology, Gliwice, Poland, in 2004 and Ph.D. degree also in electronics at Department of Electronics, Silesian University of Technology, Gliwice, Poland, in 2014. He is currently the assistant professor at Department of Automatic Control. His main research interests include the wide range of image processing issues and the technical aspects of the astronomy. Adam Popowicz is the member of the directorate of BRITE project – first polish scientific satellite mission – wherein he is responsible for denoising images acquired by the nanosatellites. He also participates in several national and international projects on astronomical and biomedical image processing, being the project leader in one of them. Adam Popowicz published several papers in refereed journal and international conference proceeding.



**Bogdan Smolka** received the Diploma in Physics degree from the Silesian University, Katowice, Poland, in 1986 and the Ph.D. degree in computer science from the Department of Automatic Control, Silesian University of Technology, Gliwice, Poland, in 1998. From 1986 to 1989, he was a Teaching Assistant at the Department of Biophysics, Silesian Medical University, Katowice, Poland. From 1992 to 1994, he was a Teaching Assistant at the Technical University of Esslingen, Germany. Since 1994, he has been with the Silesian University of Technology. In 1998, he was appointed as an Associate Professor in the Department of Automatic Control. He has also been an Associate Researcher with the Multimedia Laboratory, University of Toronto, Canada since 1999. In 2007, Dr. Smolka was promoted to Professor at the Silesian University of Technology. He has published over 200 papers on digital signal and image processing in refereed journals and conference proceedings. His current research interests include low-level color image processing, human-computer interaction and visual aspects of image quality.